

CLAIMS

WHAT IS CLAIMED IS:

1. A method of recalling data objects stored on a plurality of media comprising:

receiving a plurality of requests to recall data objects, each data object being located on a particular one of said plurality of media;

creating a plurality of queues, wherein each one of said queues corresponds to one of said media;

placing said requests on the created queues, wherein each request is placed on the queue corresponding to the medium on which the requested data object is located;

activating a first of said queues, said first queue being associated with a first of said plurality of media; and

retrieving, from said first medium, the data objects requested on the first queue.

2. The method of claim 1, wherein each queued request indicates the offset from a predetermined location on a medium at which the requested data object is located, and wherein said placing act comprises:

organizing the requests on each queue in a first and a second sequence, each sequence comprising a set of requests whose offsets are monotonically increasing within the respective sequence.

3. The method of claim 1, further comprising:

activating a second of said queues, said second queue being associated with a second of said plurality of media, the second medium being different from said first medium; and

1 retrieving, from said second medium, the data objects requested
2 on the second queue, the retrieval occurring at least in part concurrently with the
3 retrieval of data objects from the first medium.

4
5 4. The method of claim 1, wherein each of said queue comprises a linked
6 list of requests.

7
8 5. The method of claim 1, further comprising querying a database to
9 determine the locations of the requested data objects.

10
11 6. The method of claim 5, wherein the location provided by said
12 database comprises a media identifier and an offset into the identified medium.

13
14 7. The method of claim 1, wherein the act of creating a plurality of
15 queues comprises timestamping each of the created queues, and wherein the act of
16 activating a first of said plurality of queues comprises selecting a queue to be activated
17 based on the timestamps of the respective queues.

18
19 8. The method of claim 1, said method being performed in a multi-
20 threaded environment wherein plural threads execute concurrently, said method further
21 comprising:

22 acquiring a lock prior to performing an action selected from the
23 group consisting of:

24 placing a request on a queue; and
25 changing a queue from a non-active state to an active
26 state;
27 said lock being acquirable by only one of said threads at a given time; and

1 releasing said lock subsequent to performing an action in said
2 group.

3
4 9. The method of claim 1, wherein said method is performed in a
5 computing environment comprising a hard disk, said method further comprising:

6 determining that said data objects are not located on said hard
7 disk.

8
9 10. A computer-readable medium having computer-executable
10 instructions to perform the method of claim 1.

11
12 11. In a computing environment having a first plurality of drives in
13 which data objects to be retrieved are stored on a second plurality of media mountable
14 on said drives, a method of obtaining data objects from said media comprising:

15 identifying a first data object located on a first of said media;

16 identifying a second data object located on a second of said media
17 different from said first medium; and

18 concurrently using a first and a second of said drives to retrieve
19 said first and second data objects from said first and second media.

20
21 12. The method of claim 11, further comprising:

22 creating a first queue corresponding to said first medium;

23 placing first data on said first queue, said first data being
24 indicative of said first data object;

25 creating a second queue corresponding to said second medium;

26 and

27 placing second data on said second queue, said second data being
28 indicative of said second data object.

1
2 13. The method of claim 12, wherein said first data comprises a location
3 of said first data object on said first medium, and wherein said second data comprises a
4 location of said second data object on said second medium.

5
6 14. The method of claim 11, further comprising:
7 identifying a third data object located on a third of said media;
8 waiting for said first or said second data object to be retrieved
9 from their respective media; and
10 retrieving said third data object from said third medium.

11
12 15. A computer-readable medium having computer-executable
13 instructions to perform the method of claim 11.

14
15 16. A method of scheduling requests to recall data objects from a
16 medium, said medium being mounted on a drive, said drive having a reading head, said
17 method comprising:

18 receiving a new request to recall a data object from said medium,
19 said data object being located at a first offset along said medium;

20 identifying a queue comprising a plurality of requests to recall
21 data from said medium, said requests having an order, each of said requests
22 corresponding to a particular data object stored on said medium and indicating an offset
23 into said medium at which the corresponding data object is located, said requests
24 forming first and second sequences, said first sequence preceding said second sequence
25 in said queue with respect to said order, wherein the offsets of the requests within said
26 first sequence are a maximally monotonically increasing series;

27 determining the location of said reading head along said medium;

1 determining that the location of said reading head is beyond said
2 first offset;

3 inserting said new request into said queue into a position that,
4 with respect to said order, is subsequent to said first sequence.
5

6 17. The method of claim 16, wherein the offsets of the requests within
7 said second sequence are a maximally monotonically increasing series.
8

9 18. The method of claim 17, wherein the position in said second
10 sequence in which said new request is inserted is based on said first offset, such that the
11 monotonically increasing nature of said second sequence is preserved following the
12 insertion.
13

14 19. The method of claim 16, further comprising:
15 selecting, from among a plurality of queues, a particular queue
16 into which to insert said new request, wherein each one of said plurality of queues
17 corresponds to a different medium, the selection being based on the particular medium
18 on which the data object is located.
19

20 20. A computer-readable medium having computer-executable
21 instructions to perform the method of claim 16.
22

23 21. A system for retrieving data objects from a plurality of media
24 comprising:

25 a queuing module which creates a plurality of queues
26 corresponding to said plurality of media, said queuing module receiving requests to
27 retrieve data objects from said plurality of media and queuing each of said requests on
28 the queue corresponding to the medium on which the requested data object is located;

1 an activation module which selects queues for activation and
2 activates the selected queues; and

3 a retrieval module which retrieves the items on one of the
4 selected queues from the corresponding medium in the order in which the items are
5 located on the queue.

6
7 22. The system of claim 21, further comprising:

8 a hard disk;

9 a file system which manages files on said hard disk, which stores
10 information indicating which of said files have been migrated to said media, and which
11 issues a request to said queuing module for requested files that have been migrated to
12 said media.

13
14 23. The system of claim 21, wherein said queuing module timestamps
15 each queue at the time that the queue is created, and wherein said activation module
16 selects a queue for activation based on the timestamps of the created cues.

17
18 24. The system of claim 21, wherein each requested data object is
19 located at an offset into the medium on which the data object is located, and wherein
20 said queuing module includes logic which maintains the requests on each of said
21 plurality of queues in first and second sequences, the offsets of the data objects
22 requested in each of said first and second sequences comprising a maximally
23 monotonically increasing series.

24
25 25. A system for recalling data objects from a plurality of media
26 comprising:

27 means for receiving requests to recall data;

28 means for creating a plurality of queues;

- 1 means for placing the requests on said queues according to the
- 2 media on which the requested data is located;
- 3 means for activating said queues; and
- 4 means for retrieving data requested on an activated queue.
- 5

409 408 407 406 405 404 403 402 401 400 399 398 397 396 395 394 393 392 391 390 389 388 387 386 385 384 383 382 381 380 379 378 377 376 375 374 373 372 371 370 369 368 367 366 365 364 363 362 361 360 359 358 357 356 355 354 353 352 351 350 349 348 347 346 345 344 343 342 341 340 339 338 337 336 335 334 333 332 331 330 329 328 327 326 325 324 323 322 321 320 319 318 317 316 315 314 313 312 311 310 309 308 307 306 305 304 303 302 301 300 299 298 297 296 295 294 293 292 291 290 289 288 287 286 285 284 283 282 281 280 279 278 277 276 275 274 273 272 271 270 269 268 267 266 265 264 263 262 261 260 259 258 257 256 255 254 253 252 251 250 249 248 247 246 245 244 243 242 241 240 239 238 237 236 235 234 233 232 231 230 229 228 227 226 225 224 223 222 221 220 219 218 217 216 215 214 213 212 211 210 209 208 207 206 205 204 203 202 201 200 199 198 197 196 195 194 193 192 191 190 189 188 187 186 185 184 183 182 181 180 179 178 177 176 175 174 173 172 171 170 169 168 167 166 165 164 163 162 161 160 159 158 157 156 155 154 153 152 151 150 149 148 147 146 145 144 143 142 141 140 139 138 137 136 135 134 133 132 131 130 129 128 127 126 125 124 123 122 121 120 119 118 117 116 115 114 113 112 111 110 109 108 107 106 105 104 103 102 101 100 99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82 81 80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0